

TP2 Booking : Diagrammes de cas d'utilisation et de classes

Pour une application inspirée de l'application Booking à implémenter ultérieurement en Java, décrire les diagrammes de cas d'utilisation et de classes respectant les instructions suivantes **[source : aides de chatGPT]**.

⚠ **Ce TP2 Booking est à faire en binôme du même groupe de TD. Ce TP2 sera à déposer ultérieurement sur BoostCamp, selon les consignes qui vous seront mentionnées.**

Pour dessiner les diagrammes de cas d'utilisation et de classes de l'application Booking, vous pouvez utiliser le logiciel dont le lien est [draw.io \(diagrams.net\)](https://draw.io) ou équivalent

1. Sur DropBox, cliquer sur le bouton « Créer un nouveau diagramme » (Create new diagram)
2. Cliquer sur « Diagramme vierge » (Blank Diagram) et le bon « Créer » (Create)
3. Si demandé par DropBox, cliquer « Autoriser » et connectez-vous par exemple avec votre compte gmail
4. Vous disposez de plusieurs formes à gauche, mais surtout d'une option « UML » à utiliser avec d'autres formes, dont celle d'une classe (ClassName avec field et method) et plusieurs traits de relations (à vous de choisir le bon en fonction de la relation entre classes).

Pour enregistrer la progression : « Fichier > Enregistrer sous... » (File > Save as...).

Pour exporter en image, pdf, html, etc ... : « Fichier > Exporter en tant que » (File > Export as).

Exercice : diagrammes de cas d'utilisation et de classes de l'application Booking

En vous inspirant du [Cours 2 et exercices Semaine 2 Diagrammes de cas d'utilisation et de classes](#), dessinez les diagrammes des cas d'utilisation et diagrammes de classe de l'application Booking, en tenant compte des indications ci-dessous.

Vous pouvez vous aider de ChatGPT ou équivalent d'IA générative pour vous prodiguer des conseils, en citant bien vos sources et surtout en citant le pourcentage d'aide qui vous sera demandé.

➤ **Question 1 :** Définir une classe *Hébergement* avec les propriétés suivantes à titre d'exemples.

- Exemples d'attributs : l'identifiant unique, le nom de l'hébergement, l'adresse postale, le type d'hébergement (exemple : hôtel, appartement, villa...), le nombre maximal de personnes, le prix pour une nuit, la description générale du logement, le descriptif des équipements (WiFi, TV, cuisine, etc.), etc.
- Exemples de méthodes :
 - Vérifier si les dates sont libres ;
 - Retourner un résumé de l'hébergement avec toutes les informations nécessaires : son nom, son adresse, son éventuelle localisation sur la carte, ses photos, les commentaires de ses clients, son descriptif des options proposées (par exemple, le Wifi, le service ménage, la climatisation, fumeur ou non-fumeur, le coffre-fort, la vue, le petit-déjeuner compris ou pas, le parking privé, le bar, l'environnement naturel, les plages, les transports en commun, les aéroports proches etc.) etc. ;
 - Retourner le prix total du séjour ;
 - Ajouter une période disponible (entre une date d'arrivée et de départ) ;
 - Supprimer une période disponible (entre une date d'arrivée et de départ) ;
 - Ajouter une note et recalculer la moyenne des notes de cet hébergement ;
 - Etc.

➤ **Question 2 :** Définir une classe *Client* avec des attributs et méthodes communes à titre d'exemples.

- Exemples d'attributs : le nom, le prénom, l'email, le mot de passe, l'adresse, la date d'inscription etc.
- Exemples de méthodes :
 - Filtrer (chercher) des hébergements selon plusieurs critères : type d'hébergement (hôtel, maison, appartement, chambre d'hôte...), destination, dates d'arrivée et de départ, fourchette de prix, nombre

d'étoile pour un hôtel, destination depuis le centre, activités sportives et attractions, , prix, date d'arrivée etc.) ;

- Visualiser les informations ;
- Réserver un hébergement en indiquant les informations nécessaires (la destination, la date d'arrivée, le nombre d'adultes, d'enfants, de chambres etc.) ; annuler une réservation ; vérifier si le client a une réduction en fonction de sa fidélité.

➤ **Question 3 :** Définir les 2 classes suivantes spécialisées de clients héritant de la classe *Client* ci-dessus, avec les spécificités ci-dessous de ces 2 classes à titre d'exemples :

- a) *NouveauClient* : nouveau client qui doit s'inscrire avec un email et un mot de passe. Si besoin, certaines méthodes de ce nouveau client doivent être redéfinies par rapport à celles de sa super-classe du client commun. Par exemple, ce client n'a pas de réduction.
- b) *AncienClient* : ancien client qui doit se connecter avec son email et son mot de passe pour accéder à son historique des hébergements déjà réservés. Si besoin, certaines méthodes de cet ancien client doivent être redéfinies par rapport à celles de sa super-classe du client commun. Par exemple, ce client peut avoir droit à une réduction, en fonction de son nombre de réservations.

➤ **Question 4 :** Définir une classe *Réservation* qui la relie aux classes *Client*, *Hébergement* et d'une période (dates de début et de fin). Elle peut aussi contenir des informations sur le statut, le prix, des réductions, etc.

- Exemples d'attributs : l'identifiant unique, le statut de la réservation (en attente, confirmée ou annulée), l'objet du client qui a réservé, l'objet de l'hébergement réservé, les dates d'arrivée et départ, le prix total à payer (réduction incluse), la date à laquelle la réservation a été faite, etc.
- Exemple de méthodes :
 - Calculer le prix total (avec/sans réduction) en fonction du nombre de nuits (entre les dates d'arrivée et départ) ;
 - Vérifier si la réservation est confirmée et en cours (selon son statut).
 - Appliquer une réduction au prix en fonction du pourcentage de réduction ;
 - Annuler la réservation (changer son statut) ;
 - Etc.

Vous pouvez vous aider de ChatGPT, Copilot ou équivalent d'IA générative pour vous prodiguer des conseils, en citant bien vos sources et surtout en bien comprenant cette aide et en citant le pourcentage d'aide qui vous sera demandé.

N'oubliez pas de citer vos sources : auteurs, liens web, etc.